Bachelor's Thesis (UAS)

Degree Programme: Information Technology

Specialization: Information Technology

2012

Shaoyu Wang

Portsmouth City Bus Route Online Information System

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme: Information Technology | Specialization: Information Technology

June 2012 | Number of pages: 53

Instructor: Patric Granholm

Shaoyu Wang

Portsmouth City Bus Route Online Information System

City bus transport is an important component of city infrastructure, which is closely linked with travel tools. At the present time, most of people prefer to travel by bus as their first choice. However, today because of the complex bus lines, it is obvious that an online bus route inquiry may help people plan their travel and find the best bus routes.

The aim of this project is to design and implement a city bus route online information system, using Portsmouth as an example. In Portsmouth, First Group is operating a part of bus lines, and has their bus information on their web page. However, if travelers hope to take a bus from place A to place B, they need to cross-check several maps. The purpose of this project is to compile the information about bus routes and allow a point and click map interface for travelers to pick up a best route from place A to place B. The programming language used in this project to complete the functions and interface design is Javascript. The Google maps API could be considered as well in this project, which can be more efficient to implement the bus route online information system. It realizes the database connection, updates the code, thus achieves to manage the city bus route online information, such as route information add, delete, and modify. The database chosen is phpMyAdmin which creates and manages the tables, including administrator, bus station, bus information, station information, etc. The SQL language is used to achieve its database operation.

KEYWORDS:

Information system, bus route query, MySQL, HTML, PHP, Javascript

FOREWORD

This thesis is the last task for my bachelor study. During these four years, my teachers and classmates are like my family members to friendly get together with me. We learn each other, help each other, and grow up together. I am really grateful to them. Especially for my thesis supervisor Mr. Patric Granholm, as my thesis could not be completed smoothly without his help, as well as my tutor Mrs. Poppy Skarli, she also taught and helped me a lot during these four years.

I have to say that Dr. Linda Yang was very helpful for me, when I did this individual project for this thesis during my exchange study at University of Portsmouth. She supervised and encouraged me to complete the project. With her excellent teaching, I learned a lot from her.

In addition, my parents always give me endless love and care in my study abroad. So please allow me to sincerely appreciate what they do.

May 2012 Turku

Shaoyu Wang

CONTENTS

FIGURES

TABLES

1 Introduction

City public transport is closely linked with city infrastructure. At the present time, it is the tool of choice for most people travelling. Considering the complexities of in the bus lines today, the establishment of a network query system can help people rationalize travel, learn to change bus routes and facilitate access of information of various bus lines.

When local people or travellers try to find how to get from place A to place B by bus in Portsmouth, the most convenient way is to check online. This individual project is to develop and design Portsmouth city bus route planner. Through it, the users can pick up a best route. The required bus changes on the selected route should be displayed, as well. In order to complete this individual project, key technologies such as HTML, web programming, Database are used.

1.1 Background

With the innovation of the traffic tools, distance is no longer trouble for people travelling. The large and complex travelling network can be covering almost every corner of the world; so that people may reach any where they want to go. However, when people face this complicated transportation network, they may feel confused or overwhelmed. Especially in everyday life, most people prefer choosing bus to go somewhere in their city. However the bus lines are very complicated.

In this project, the Portsmouth City Bus was chosen as an example, and information about the bus route for First Group in Portsmouth City was compiled. Portsmouth city bus transport is carried out by First Group, which has already put all their bus information on their website. But if passengers want to travel from place A to place B, they may need to cross-check several maps and timetables. The purpose of this project will improve it, and allow a point and click map interface for passengers to pick the best route from place A to place B. The required changes on the selected route should be displayed by the bus timetable as well.

1.2 Objectives

The project objectives were to be complete the function and the design of interface, and realize the query for Portsmouth bus lines, including query by line, query by stop, query by start stop and end stop, provide the shortest route querying from start stop to end stop, change line query, and also support blur query. The database used was the SQL Server 2005, and the application accesses the database and uses SQL language to operate the database. Under the project developing, the system will be evaluated and improved, if it is required.

1.2.1 Collecting bus route information and building a database regarding different routes

The purpose of this individual project is for Portsmouth city bus route, and the bus routes operated by First Group is the important researched object. Through their website, there is a map that could be downloaded as PDF format. This map illustrates the whole local bus information in Portsmouth, including service No., bus route, bus station name, and bus service time.

Considered with the local bus information, the query system needs to combine the Google map with this information, so that through the system, the users can see the bus information distinctly based on Google map.

1.2.2 Developing an algorithm to find the shortest/best route

The system is not only to check the bus route, but also to find the shortest or best route for passengers. To reach this goal, the algorithm, which can calculate the shortest or best way, should be found, for example, how to find the way with the minimum transfer times, how to find the way with using the shortest time, how to find the method can pay less bus fares. This algorithm is crucial for this individual project.

The algorithm of shortest path contains the static shortest path algorithm and the dynamic shortest path algorithm. The former one is to calculate the shortest path without external environment changes. This main algorithm is Dijkstra Algorithm, A* (A Star) Algorithm [1]. The latter one is the opposite, which is to calculate the shortest path with external environment changes. This means that the shortest can be found

without predicted calculation situations, like the obstacles continuing to move. It is represented by the D* (D Star) Algorithm.

However, the most classical algorithm is Dijkstra Algorithm, which is widely used to solve the system problem about the shortest path. But for different systems, the Dijkstra Algorithm is achieved with different methods.

1.2.3 Producing an interactive map to allow users choose the best route online

Through route analysis route from place A to place B, different users may choose the way based on different requirements, the shortest way, the minimum time used, and the cheapest fare. Meanwhile, when user has long journey, the time of transfer should be considered. Someone may prefer fewer transfers, but someone may like the shortest way without considering the number of transfers.

Then how do we deal with this kind of problems? This may depend on the choices of the users. A better way will be to produce an interactive map, so that users could choose the best route online based on their own preferences.

1.2.4 Evaluating the system

The evaluation of system is to proceed testing every module. After modifying and improving, it needs to be evaluated totally. The environment of evaluation is the: Windows 7 operation system and the IE browser. The modules of evaluation are: the users query module, the administrator management module, and the feedback module. Then the results of evaluating should be analysed.

The aspects of evaluating are following:

- Evaluating the user module
- Evaluating the administrator module
- Evaluating the feedback module

2 Theory Basis

2.1 Database

A database organises, stores and manages data according to the data structure [2]. However, the data management is not only just storing and managing the data, but also managing all kinds of data that users need. The database has a variety of types, which is from the simple data storage to mass data storage systems in all respects, as well as a wide range of applications.

For example, an enterprise or institution of the personnel department will always put their own employees' information and documents (employee's ID, name, gender, age, salary, CV, etc.) in a table. This list can be seen as a database. With this "data storage", people can query the employee information they need. If these tasks can be done by computer automatically, the enterprise personnel management can achieve a higher level of efficiency. In addition, in the financial management, store management, production management, as well as bus information management, numerous database systems will be required to be built, in order to realize automatic management.

2.1.1 Database Basic Structure

There are three layers for the database basic structure, which reflect the observation of database from three different views.

1. Physical Data Layer

It is the innermost layer of the database, and a collection of the actual data stored in physical storage devices. These data are raw data, the user processes an object, composed of bit strings, character and word of the instructions described in the operation processing by the internal mode.

2. Concept Data Layer

It is the middle layer of the database, is a logical representation of the database as a whole, and points out the logical connections between the logic of each data definition and data collection of records stored. It reveals the logical relationship of all objects in

the database, rather than their physical situation. It is the concept of a data base administrator database.

3. Logical Data Layer

It is the database which the users see and use, indicating one or a number of specific user data collections, namely a collection of logical records.

The different layers of database are linked by mapping conversion.

2.1.2 Database Creating

MySQL is a kind of a connection database management system. The database will use the connection to be stored in different data storages, not all data in one data storage [3]. This can increase the speed and improving the flexibility a database. MySQL is the SQL “structured query language”, which is used to access the database of the most commonly used standardized language.GPL (the GNU general public license) is used by MySQL software. Because of its small size, quick speed, and low cost, especially the published open source, many medium or small company websites prefer to choose MySQL as their website database, in order to save the expenditure.

Considering this project, we chose to use phpMyAdmin, which is a MySQL database systems management process procedure written in the PHP language. According to it, managers can use the Web interface to manage the MySQL database by some common operation, including database management, table, field name, index, relationship, user, etc., as well as executing any SQL statements directly.

Below is the output of how a database with three fields will appear. (Figure 2.1.2)

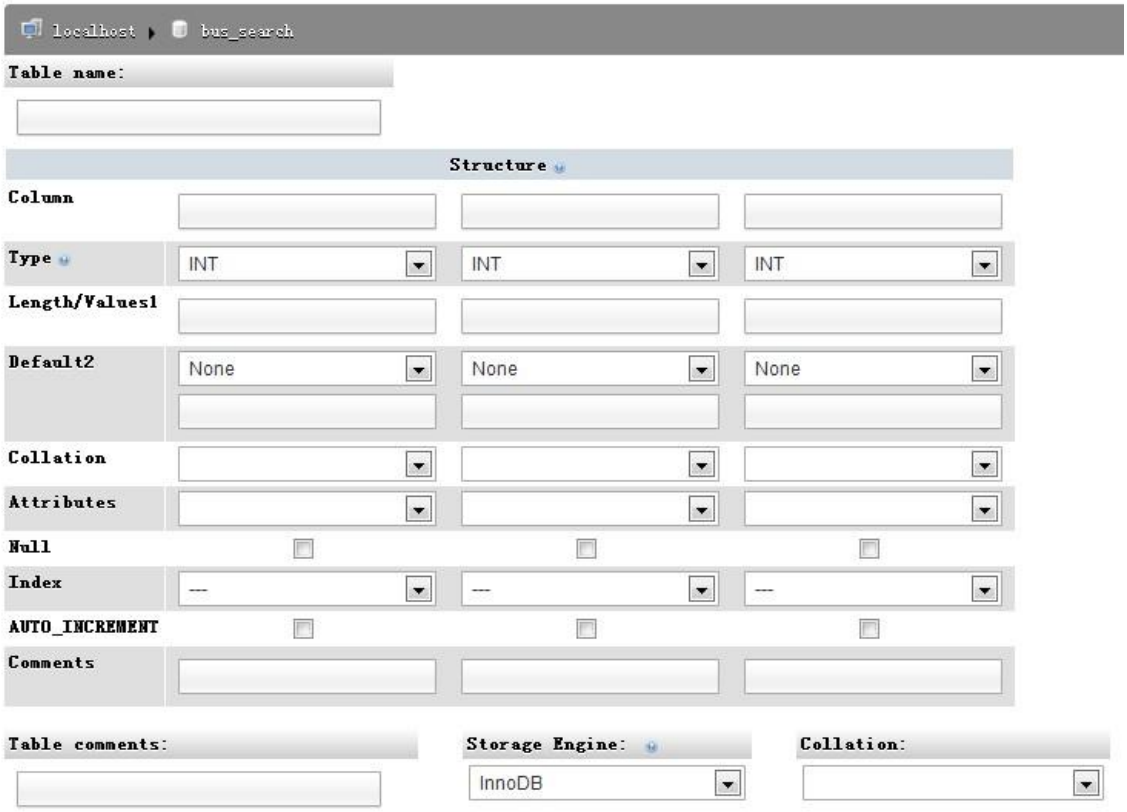

Figure 2.1.2. Database with Three Fields

2.2 HTML

HTML (HyperText Markup Language) is a main markup language used to describe web documents [4]. The format of the web page is HTML, using a combination of other web technologies, such as the scripting language, the CGI, components, etc. It can create a powerful web. Thus, HTML, web programming, that is the World Wide Web, is built on the basis of hypertext.

HTML document production is not very complex, and powerfully supports to insert the file into different data formats, the reason of which is the prevalent for WWW. Its main features are as follows:

1. Simplification

HTML version upgrading uses the superset method, and is more flexible.

2. Scalability

The HTML language is widely used to bring enhancements to increase the identifier requirements. HTML takes elements of the sub-class way to bring assurance for system expansion.

3. Independence

Despite the popularity of the PC, a lot of people still use MAC and other machines. HTML can be used in a wide range of platforms. That is another reason why it is also prevalent in the World Wide Web.

2.3 PHP

The syntax of PHP is a unique mix of C, Java, Perl, and PHP's own syntax [5]. It can be faster than the CGI or Perl implementation of dynamic web pages. Comparing dynamic web pages by PHP with other programming languages, PHP is the program embedded in the HTML document to execute higher performance than the fully CGI to generate HTML tags. PHP can also compile the code; the compiler can encrypt and optimize the code to run, and make the code run faster. It has a very powerful feature that not only can achieve all functions of CGI, but also support almost all popular database and operating system.

The main features of PHP are listed below:

1. An open source code

All PHP source code can be obtained in fact.

2. PHP is free.

Compared to other technologies, PHP is free of charge.

3. The speed is quick.

The development of application is fast, run fast, and technology learning is fast, embedded in HTML because PHP can be embedded in the HTML language. Compared with other languages, and has some advantages, such as editing and being simple, practical, and more suitable for beginners.

4. Cross-platform

Because PHP is running on the server side script, and can also be running on UNIX, LINUX, and WINDOWS.

5. High efficiency

PHP consumes few of the system resources.

6. Image processing

PHP is used to create dynamic images.

7. Object-oriented

For PHP 4 and PHP 5, the object-oriented has a great improvement. And now PHP can be used to develop large scale business processes.

8. The professional focus

PHP is based to support a scripting language, similar to the C language.

2.4 Javascript

Javascript is a programming language to make the website more lively [6]. The current page design is easy to learn, and it is also the most convenient language. A welcome message is easily made by using Javascript, as well as a beautiful digital clock, and a simple election with advertising effectiveness. It can also be used to display the stay time of browsing the device. These special effects can improve the observability of the web pages.

However, there are several differences between Javascript and Java.

1. Javascript is a dynamic, weakly typed, prototype-based language, through the browser executing directly; Java is an object-oriented programming language. It must be implemented for compilation and connection before first action.
2. Javascript can be compiled in the HTML file, which will directly view the page's source, and see the Javascript code. Then there is no protection, and anyone can copy the program through HTML files; the Java application program of the page is called Java applets, and is separate from the with HTML file.

3. The structure of Javascript is freer and looser, while the structure of Java is rigorous that is the same with other traditional programming languages.
4. Javascript can not read and write files or network control functions, but Java can provide these features. However, Javascript is used in control of web content and interactive aspects more conveniently and efficiently.
5. Javascript can always run on the client; Java can often run on the server.

3 System Requirement Analysis

This chapter deals with the system requirement analysis, including performance requirements, function requirements, and modules partitioning.

3.1 Performance Requirement Analysis

In order to guarantee that the system will have a long-term, safe, stable, reliable, efficient operation, Portsmouth city bus database should meet the following bus inquiries performance requirements:

1. System of processing accuracy, timeliness and response speed

This query system should ensure the integrity rate and all the corresponding fields should contain the records of query keywords which need to be found. In the system design and development process, it is necessary to give full consideration to the current system and to ensure that it will withstand the workload of the system's processing power and response time to meet the demand for information processing by the system administrator. Response time, update processing time should be relatively rapid, and should meet the requirements of most users. The normal operation of the response time should be within 1-2s. Data import and export operation should be completed within an acceptable time, in principle, to ensure that the operator does not affect efficiency because of speed issues.

2. System openness and scalability

In the development process of system, the scalability in the future should be fully considered. For example, user queries demand will continue to update and improve. This requires the system to provide adequate methods for the adjustment and expansion of the function. To achieve this, the system should be an open system, as long as it meets certain specifications, for example, simply joining and reducing the modules of system, as well as configuring the system hardware. The system is completed with upgrades and improvement through software patches and replacement.

3. System usability and maintenance

The system is directly used by personnel, while the user is often not very familiar with computers. This requires the system to provide an interactive and user-friendly interface. Therefore, this should be taken into account in system development. The users only need to know the URL of this system, and use directly the query module of the system without user registration and login, saving time for the user query. Moreover, to achieve the easy use of the system, it is required that the system should try to make the user familiar with the terminology and interface. It can provide online help desk, that is, set up a special “Contact” allowing the user to solve the problems, as well as designers to know the inadequacies of this system to make it more comprehensive.

3.2 Function Requirement Analysis

The system uses structured design methods to achieve the overall system function, improve the system of indicators. It means that the whole system is divided into various functional modules correctly between modules. And the module within the contact and database, it defines internal structure of each module to achieve the function of the system as a whole, that is, the relationship between the module design and module system.

3.2.1 Ordinary User Requirements

Based on the convenience of people travelling by city bus, this system needs to satisfy the following ordinary user requirements:

1. Line Query: it is convenient to find the fastest and latest line information, such as which bus stops the line is passing, and each bus stop location of this line.
2. Station Query: If user is not clear with the bus line and only knows the destination, then the station query will help user identify quickly which bus lines can take them there, but also provide the relevant information for each line, and indicate the station in the corresponding line, in order to facilitate the user to understand the location of the station on the bus line.
3. Transfer Query: If there is no direct line, the user can find the shortest bus router interchange. By entering the starting station and the terminal as the query keyword, it could be checked out how to take bus within three times transferring to the destination address. It can save more time for the user, as well as improve the efficiency of planning.
4. Saving ticket fare Query: If user prefers to choose the cheapest price to destination, the system could automatically calculate the prices of the whole travel route for different travel route options, so that the user can make a choice.

The user can process the line query, station query, transfer query, and price query. The analysis of ordinary users is shown as follows. (Figure 3.2.1)

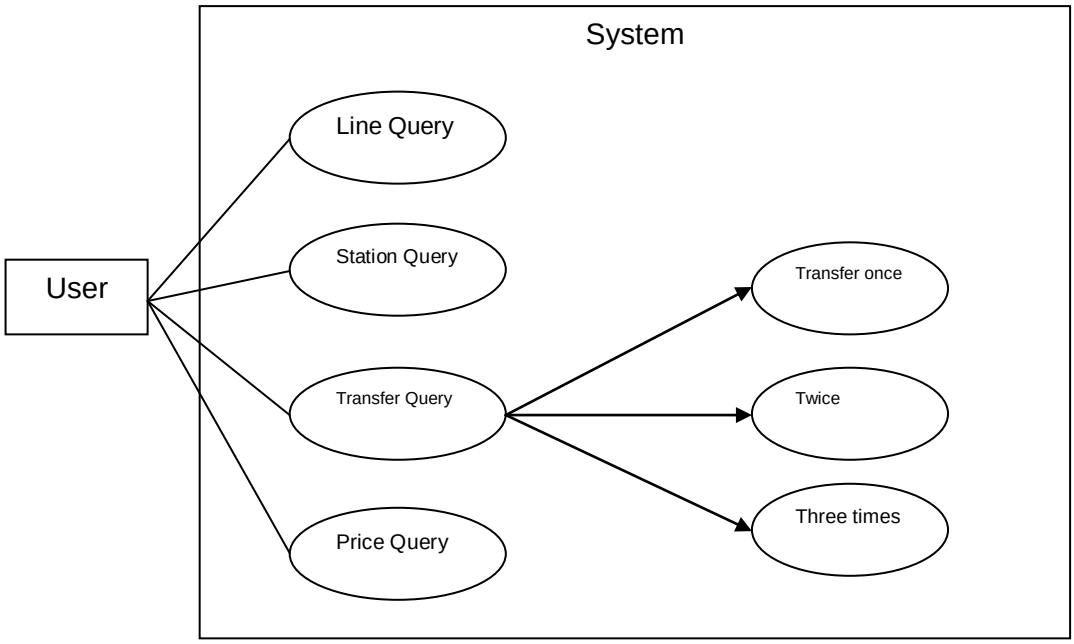

Figure 3.2.1. Analysis of User Query

3.2.2 Administrator Requirements

In order to maintain the normal operation of this system, it needs to provide background management functions for the administrator login, add/modify/delete bus data, modify the information, a secure password, reply to user messages, whose main function is to add/modify/delete data and information to ensure that the bus line is properly available. The administrator requirement analysis is shown below. (Figure 3.2.2)

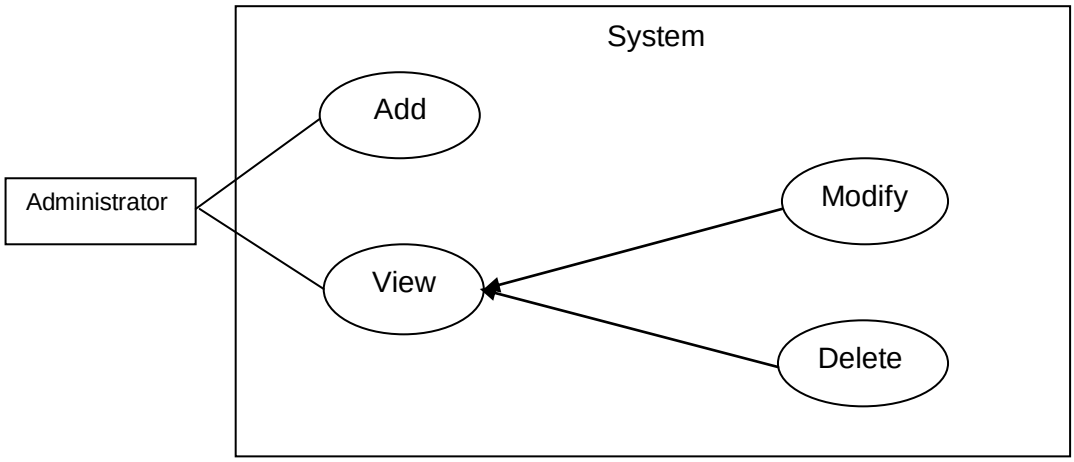

Figure 3.2.2. Analysis of Administrator Requirement

3.3 Module Partitioning

1. Ordinary User Module

This module can realize the bus query function, which searches different route options through line query, station query, start to end stop query, and price query.

2. Administrator Module

This module can realize data addition, modification, and deletion.

Module partitioning is shown below. (Figure 3.3)

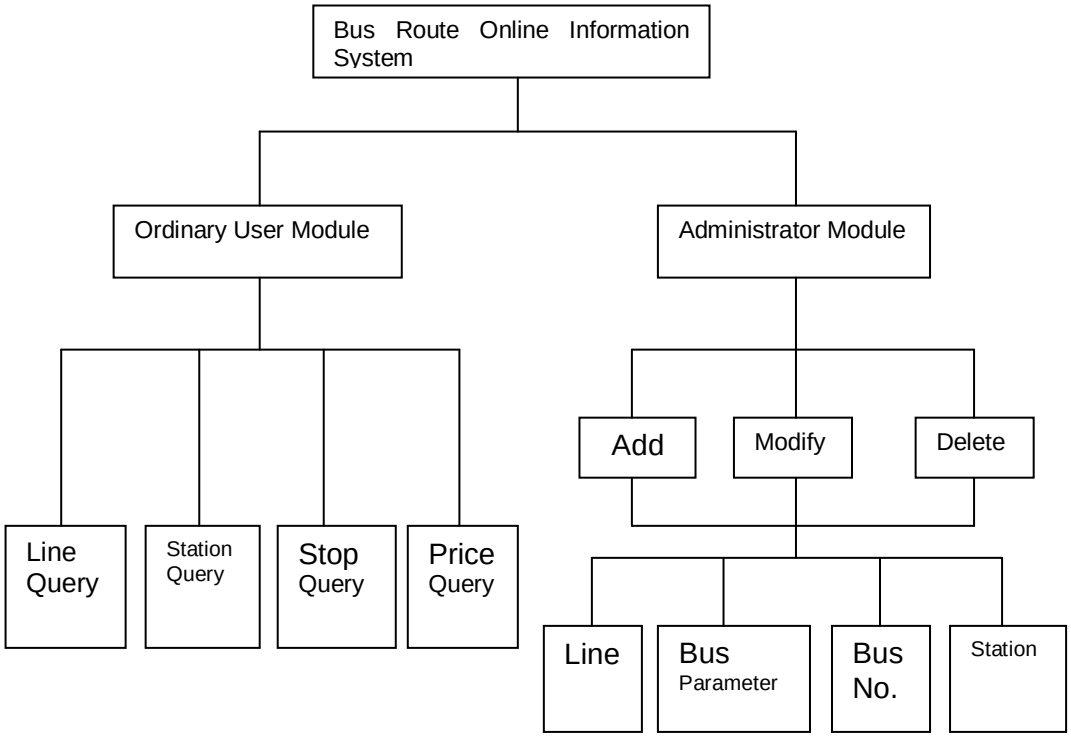

Figure 3.3. Functions of Modules

The functional modules of the system are divided as shown in the figure above, according to different functions, such as a normal user module line query, station query, and other functions. Meanwhile, the administrator module has data modification, deletion and other functions.

4 Design

4.1 Database Design

Database Design is an important part of the information system design, whose quality is directly affecting the successful implementation of the information system, the quality of the system, efficiency of the system, and the maintainability of the system.

For this system design, we should consider the multifarious management databases, and more repeatability. The databases need to be used frequently, as well. Consequently, this information system will use the MySQL Server 5.5 databases

system, which is an environment of correctly reflecting users, can accept the current system, is to easy maintain, and manages databases efficiently.

The main problem of database design is how to build a good data model. The data model is the formation of database structure links between database records. Different data management systems have different data models. In the current database management systems, there are three kinds of data models, the hierarchical model, the network model, and the relational model. Among them, the relational model has high data independence and more convenient use. This project design will use the relational database. The implementation of database adding, deleting, editing, statistics, display and print will be extremely convenient, as well as provide favourable conditions for the bus information queries.

4.1.1 Structure of Databases Conception

This system will use the Entity-Relationship Diagram (E-R Diagram) to describe the database structures and semantics, in order to simulate the real world. The E-R Diagram is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion [7]. It has two distinct advantages: it is close to people thinking, simple to understand, has nothing to do with computer, and is easy to accept by users.

The E-R Diagram is a tool to describe the concept model directly. It contains three basic parts.

1. A rectangular box, which means the type of entity someone is considering with the problem.
2. A diamond-shaped box, which means the type of relationship between entities.
3. An oval-shaped box, which means the property of entity.

The following are definitions of entity and property:

Administrator list (user name, password)

Station information list (station number, station name)

Bus information list (Bus number, start and terminal stop, time, price)

The relationship between entities is shown in Figure 4.1.1.

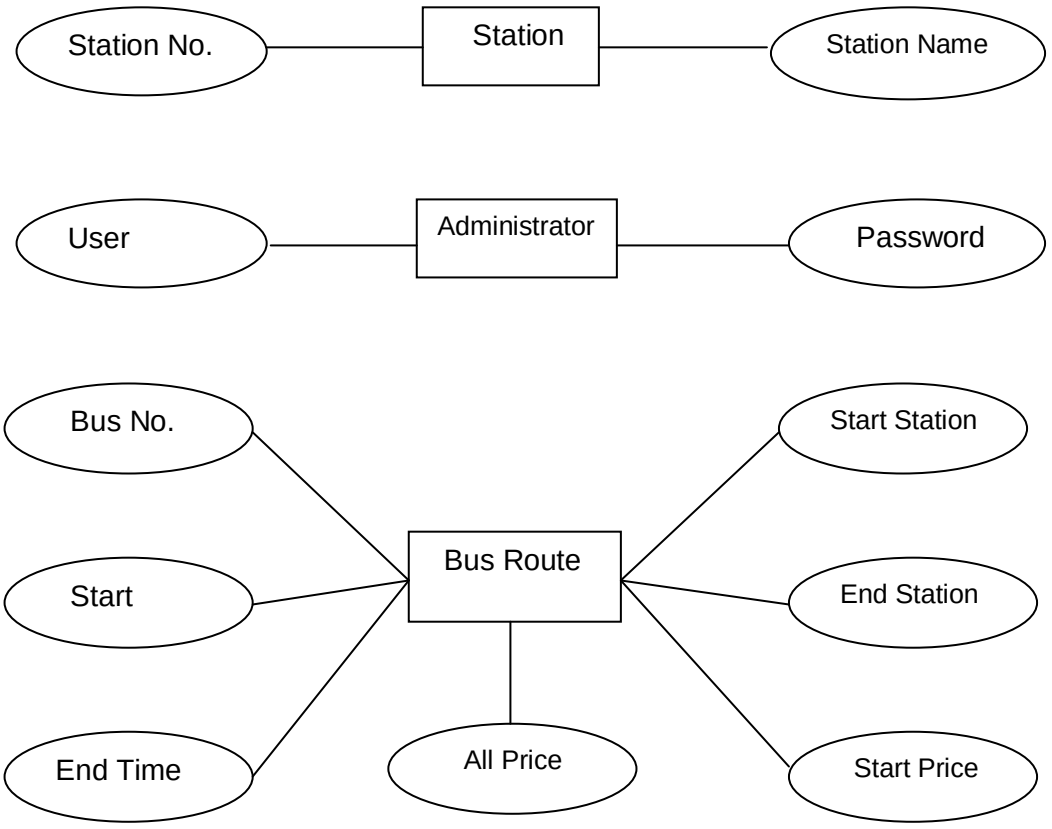

Figure 4.1.1. Relationship between entities

The draw E-R diagram helps to understand the relationship between every type of database. According to the diagram, there are three entities: station, administrator, and bus route. They have separated properties in order to prepare for the database forms.

4.1.2 Design of Databases Table

At first, databases of “Bus_search” need to be created by MySQL Server 5.5. Then every table needs to be created inside. Their structure and type are as following:

1. Administrator Table (admin)

The administrator table for login the system needs user name and password, and as well as accessing the same table while resetting a password. (Table 4.1.2.1)

Table 4.1.2.1. Administrator Table

Name	Field Name	Type	Primary Key	Null
ID	id	int	Yes	No
Username	username	varchar	No	No
Password	password	varchar	No	No
Email	email	varchar	No	No
Create Time	CreateTime	datetime	No	Yes
Update Time	UpdateTime	datetime	No	Yes

User Name: every administrator has their own user ID; user name can not contain more than ten characters.

Password: password can not be vacant and copied.

Email: the contact way of user can not be vacant.

Create Time: the time of data creating.

Update Time: the time of data updating.

2. Bus Station Table (bus_station)

The bus station table stores the station number and bus number. When the relationship between station and bus number need to be checked, the table is able to access that information. (Table 4.1.2.2)

Table 4.1.2.2. Relationship Table

Name	Field Name	Type	Primary Key	Null
ID	id	int	Yes	No
Bus ID	BusID	varchar	No	No
Station Name	StationName	varchar	No	No
Station No.	SN	int	No	No
Run Time	runtime	time	No	No

Bus ID: stands for different bus, should be numbers.

Station Name: stands for different station and should be the name of every station.

Station No.: the order of stations for one bus line.

Run Time: the time of bus to run from start stop to end stop.

3. Bus Information Table (businfo)

Bus Information Table is to put on the bus name information. When bus name needs to be found, the information could be checked through this table. (Table 4.1.2.3)

Table 4.1.2.3. Bus Information Table

Name	Field Name	Types	Primary Key	Null
ID	id	int	Yes	No
Bus Name	BusName	varchar	No	No
Start Time	StartTime	datetime	No	No
End Time	EndTime	datetime	No	No
Bus Introduction	BusIntro	varchar	No	No
Create Time	CreateTime	datetime	No	Yes
Update Time	UpdateTime	datetime	No	Yes

Bus Name: stands for different buses and should be the name of every bus.

Start Time: the time of bus starting from the first stop.

End Time: the time of bus ending to the last stop.

Bus Introduction: introduction of every bus line.

4. Station Information Table (stationinfo)

The station Information Table is to save the station information. It may be visited when station information needs to be checked. (Table 4.1.2.4)

Table 4.1.2.4. Station Information Table

Name	Field Name	Type	Primary Key	Null
ID	id	int	Yes	No
Station Name	StationName	varchar	No	No
X float	Xlat	varchar	No	No
Y float	Ylat	varchar	No	No
XY	XY	varchar	No	No
Station Introduction	StationIntro	varchar	No	No
Create Time	CreateTime	datetime	No	Yes
Update Time	UpdateTime	datetime	No	Yes

Station Name: it is to store every bus stop name.

X/Y float: the location of every bus stop.

Station Introduction: the bus stop which buses pass by

In order to ensure the independence of every table and reduce the relationship, the design of database table will be an independent, and store independent information on their own, it will be easy to operate with few error.

4.2 Modules Design

This chapter is an introduction of the main modules design, which contains the ordinary user modules design and the administrator modules design.

4.2.1 Ordinary User Modules Design

The query module is the most important module. In this module, users can query bus information without login. They can use the system to do bus number query, station name query, station to station query, and so on. Every performance is crucial. The flow chart is shown as following (Figure 4.2.1).

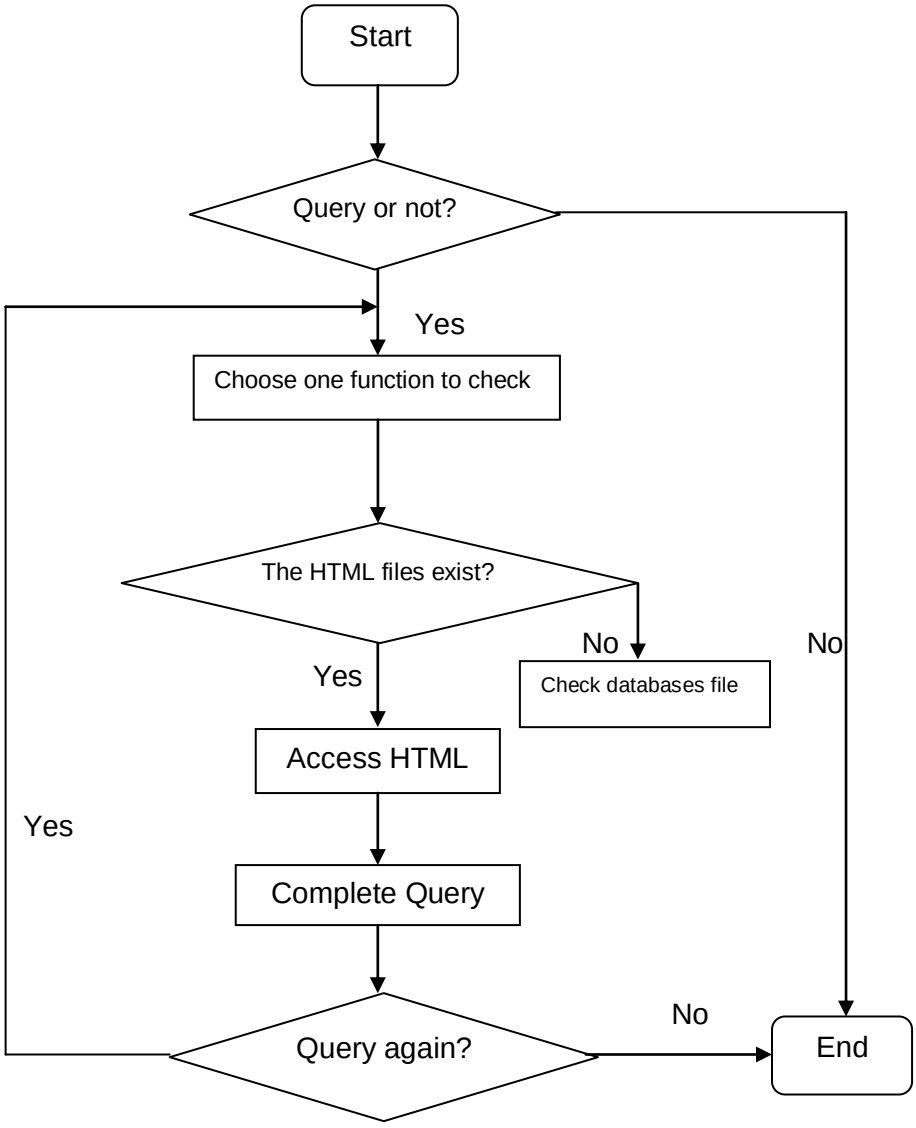

Figure 4.2.1. Query Procedure Flow Chart

In the query modules, the burden of the system can be reduced if we access the corresponding HTML files, which is an important step of query.

4.2.2 Administrator Modules Design

The administrator module is a crucial module, which is needed to prevent someone from breaking or modifying the system as the administrator login. However, the true administrator can login, and manage the databases to maintain the normal operation. The following is the flow chart (Figure 4.2.2).

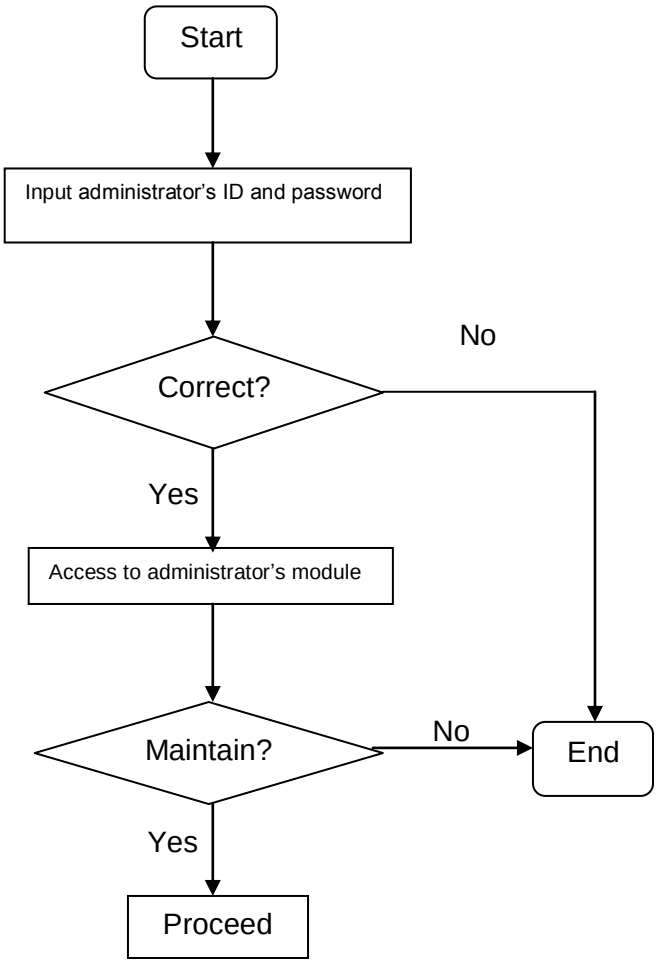

Figure 4.2.2. Administrator Procedure Flow Chart

These two modules realize their own different functions, and each of them has clear authority and responsibility, and is easy to understand.

5 Implementation

5.1 Google Maps API

Google maps API is a way in which a Google map is embedded in the web page API through Javascript. The API can provide a large number of practical tools for dealing with the map (as http://maps.google.com web site map), and through the various service adding content to the map, it will let the web site create comprehensive functions of maps application.

A simple example of Google maps API is shown as follows (Table 5.1.1).

Table 5.1.1. Google Maps API Code

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no"/>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>BUS</title>

<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
  var directionDisplay;
  var directionsService = new google.maps.DirectionsService();
  var map;
  var haight = new google.maps.LatLng(<?php echo $fromaddress ?>);
  var oceanBeach = new google.maps.LatLng(<?php echo $toaddress ?>);
```

The URL https://maps.googleapis.com/maps/api/js is to point out the location of Javascript file, the document will load to use all the symbols and definitions from the Google maps API. The web page must be including the script tags of the website.

Because the bus started stop and end stop will be the specific points, a LatLng value needs to be created in order to save these location information which will be passed to the map of the options. These two addresses will be obtained from the table of X-Y coordinate.

Table 5.1.2. Google Maps Function Initialize

```
function initialize() {
  directionsDisplay = new google.maps.DirectionsRenderer();
  var myOptions = {
    zoom: 14,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    center: haight
  }
  map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
  directionsDisplay.setMap(map);
   calcRoute();
}

function calcRoute() {
  var selectedMode ="DRIVING";// document.getElementById("mode").value;
  var request = {
      origin: haight,
      destination: oceanBeach,
      // Note that Javascript allows us to access the constant
      // using square brackets and a string value as its
      // "property."
      travelMode: google.maps.DirectionsTravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(response);
    }
  });
}
</script>
```

The Google map function initialize and function calculate routes are shown as the codes above. (Table 5.1.2) The initial zoom level is set, and the mapTypeId is set to google.maps.MapTypeId.ROADMAP. ROADMAP is used to show the default ordinary two-dimensional figure pieces.

5.2 Bus Stops

As the design plan, the database of bus stop information is created by phpMyAdmin. The structure of the bus stop information is shown below (Figure 5.2.1).

localhost ▸ bus_search ▸ stationinfo

Browse | Structure | SQL | Search | Insert | Export | Import | Operations

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None		Change Drop More
2	StationName	varchar(100)	latin1_swedish_ci		No	None		Change Drop More
3	Xlat	varchar(20)	latin1_swedish_ci		No	None		Change Drop More
4	Ylat	varchar(20)	latin1_swedish_ci		No	None		Change Drop More
5	XY	varchar(50)	latin1_swedish_ci		No	None		Change Drop More
6	StationIntro	varchar(500)	latin1_swedish_ci		No	None		Change Drop More
7	CreateTime	datetime			Yes	NULL		Change Drop More
8	UpdateTime	datetime			Yes	NULL		Change Drop More

Figure 5.2.1. Bus Stop

All of the bus stops are present according to this table. One bus stop name will be related with one location, which is the X-Y coordinate. Station introduction states which bus will pass by the certain stops. Create time and update time is illustrated when the administrator creates and modifies the table of bus stop information.

5.3 Bus Time

The calculation for the time taken suffers the problems that time is generated by adding up the time for every individual station and presenting them for the user. At first, the data of bus time was chosen as “time” but adding two times together was not simple as 00:00:00 + 00:00:00 but it was more complicated. Therefore the time data is changed to an integer making any calculations a lot simpler.

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None		Change Drop More
2	BusID	int(11)			No	None		Change Drop More
3	StationName	varchar(50)	latin1_swedish_ci		No	None		Change Drop More
4	SN	int(11)			No	None		Change Drop More
5	runtime	time			No	None		Change Drop More
6	ticketPrice	int(11)			No	None		Change Drop More

Figure 5.3.1. Before Time Format Changed

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None		Change Drop More
2	BusID	varchar(50)	latin1_swedish_ci		No	None		Change Drop More
3	StationName	varchar(50)	latin1_swedish_ci		No	None		Change Drop More
4	SN	int(11)			No	None		Change Drop More
5	runtime	int(11)			No	None		Change Drop More
6	ticketPrice	decimal(10,0)			No	None		Change Drop More

Figure 5.3.2. After Time Format Changed

The run time is shown as the time between two bus stations. The unit of time is minutes. An example is marked through the screenshot below (Figure 5.3.3).

	ID	BusID	StationName	SN	runtime	ticketPrice
Edit Inline Edit Copy Delete	1	6	Hard Interchange	1	3	2
Edit Inline Edit Copy Delete	2	6	Palmerston Road	2	7	2
Edit Inline Edit Copy Delete	3	6	South Parade Pier	3	4	2

Figure 5.3.3. Run time

When the user chooses the departure and destination, the program will calculate the sum of the every run time among all stations passing.

In addition, the bus start time and end time will be also stated through the bus information table (Figure 5.3.4).

localhost ▸ bus_search ▸ businfo

Browse | Structure | SQL | Search | Insert | Export | Import | Operations

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None		Change Drop More
2	BusName	varchar(100)	latin1_swedish_ci		No	None		Change Drop More
3	StartTime	int(11)			No	None		Change Drop More
4	EndTime	int(11)			No	None		Change Drop More
5	BusIntro	varchar(100)	latin1_swedish_ci		No	None		Change Drop More
6	CreateTime	datetime			Yes	NULL		Change Drop More
7	UpdateTime	datetime			Yes	NULL		Change Drop More

Figure 5.3.4. Start Time and End Time

Consequently, the bus time can be obtained from these tables. When the users find how to travel from place A to place B by bus, they will have the bus time information at the same time.

5.4 Bus Ticket Price

The data of bus ticket price is handled similarly to the bus time as shown the table below (Figure 5.4.1).

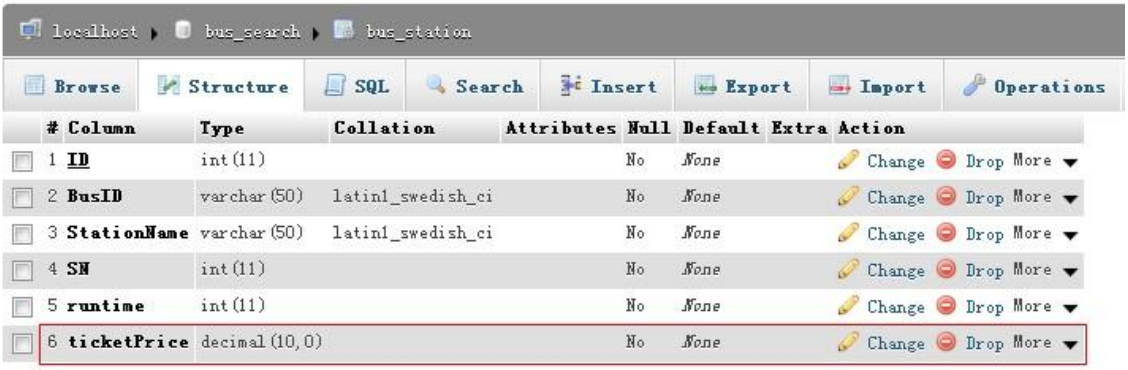

Figure 5.4.1. Bus Ticket Price

When the query of bus search is given, the result will display the bus number, as well as the bus time and ticket price.

5.5 Connecting to the Database

In a dynamic website, the use of database is quite frequent. In order to avoid a repeat of the code written, it is very important to compile a database connection. The file contains the system of database connection code. It is shown as follows (Table 5.5.1).

Table 5.5.1. Database Connection Code

```
$connect=MySQL_connect('localhost','root','123') or die ("!error connect");
MySQL_select_db("bus_search",$connect);
```

The MySQL database user is the root, and the password is 123. This database name is called bus_search. The database can be connected through the code above.

Table 5.5.2. Location of Station

```
$myquery=MYSQL_query("select * from stationinfo where XY='{$_POST['fromaddress']}'",$connect);
if(@mysql_num_rows($myquery)>0)
{
$fromaddressName= mysql_result($myquery,0,"StationName");
}
//$sql='select * from stationinfo where XY=':{$_POST['fromaddress']}'';
//echo $sql;
$myEndquery=MYSQL_query("select * from stationinfo where XY='{$_POST['toaddress']}'",$connect);
if(@mysql_num_rows($myEndquery)>0)
{
$ToaddressName= mysql_result($myEndquery,0,"StationName");
}
//echo $myEndquery;
```

The X-Y coordinates ensure the specific location of the station. After fault tolerance and the processing, then the data execute to define the variable values, and save the start station and end station.

Table 5.5.3. Bus Line

```
$myEndqueryForLine=MYSQL_query("SELECT a.busid
FROM (

SELECT *
FROM bus_station
WHERE StationName = '{$fromaddressName}'
)a, (

SELECT *
FROM bus_station
WHERE StationName = '{$ToaddressName}'
)b
WHERE a.BusID = b.BusID ",$connect);
```

The table above shows how to obtain the name of the bus line (Table 5.5.3).

Table 5.5.4. Sum of Bus Run Time

```
$myEndquery=MYSQL_query("SELECT  SUM(runtime) AS runtime  FROM bus_station WHERE busid='{$BusID}' ORDER BY sn asc",$connect);
if(@mysql_num_rows($myEndquery)>0)
{
$runtime= mysql_result($myEndquery,0,"runtime");
}
$ticketPricequery=MYSQL_query("SELECT ticketPrice  FROM bus_station WHERE busid='{$BusID}' ORDER BY sn asc  limit 1
",$connect);
if(@mysql_num_rows($ticketPricequery)>0)
{
$ticketPrice= mysql_result($ticketPricequery,0,"ticketPrice");
}
```

The sum of the bus run time can be calculated. And at the same time, the bus ticket price can be shown according to the related bus ID. (Table 5.5.4)

Table 5.5.5. Bus Transfer Code

```
$queryFortransfer=MYSQL_query("select distinct a.stationName from
(
select * from  `bus_station` where busid in(
SELECT
  `BusID`
FROM
  `bus_station` where stationName='{$fromaddressName}' or stationName='{$ToaddressName}')
  ) a,
  (
select * from  `bus_station` where busid in(
SELECT
  b1.`BusID`
FROM
  `bus_station` b1, `bus_station` b2
 where b1.busid = b2.busid and b1.stationName='{$fromaddressName}' and b2.stationName='{$ToaddressName}')
  ) b where a.stationName=b.stationName and a.busid!=b.busid",$connect);
```

Through the code above, the user can obtain get the bus station name for the bus transfer. The method for the program is firstly to list all the stations of bus line 1, as well as bus line 2. Then it can choose the same station C, so that it can act as the transfer place. Therefore, the transfer from A to B can be completed by A to C, then C to B. (Table 5.5.5)

Table 5.5.6. Transfer Information

```
$Forquerytransfer =MySQL_query("select * from  `bus_station` where busid in(
SELECT
  b1.`BusID`
FROM
  `bus_station` b1, `bus_station` b2
 where b1.busid = b2.busid and b1.stationName='{$fromaddressName}' and b2.stationName='{$ToaddressName}') ORDER BY id asc",$connect);
```

Finally, the transfer information suggestion can be shown (Table 5.5.6).

6 Testing and Evaluation

In order to ensure that this system can run normally, it is necessary to carry out a comprehensive test after completion.

The application system should be the alternation between creating and testing. It should pay attention to developing efficiency, as well as its stability. After writing each module, the module needs to be tested and evaluated, so that we can see if it can be working according to the specific requirements. The earlier the problems are detected, the earlier the problems can be solved. Otherwise the problems may greatly increase if tested in the end.

6.1 The Results of Output

To demonstrate the results, there are two different routes of output, one without connections and one with connections.

6.1.1 Output without connections

Route 1 – From Hard Interchange to Baffins, Portsmouth College by Bus No. 14 only.

Depart:Hard Interchange
Arrive: Baffins, Portsmouth College
Take Bus No.14 Line.

Expected Time: 27 min
Ticket Price: 2 Pounds

Bus No.14 Stations Information:

ID	Bus No.	Station Name	Run Time	Ticket Price
21	14	Hard Interchange	3 min	2 Pounds
22	14	City Centre	5 min	2 Pounds
23	14	Lake Road/Kingston Road	7 min	2 Pounds
24	14	Copnor Bridge	4 min	2 Pounds
25	14	Baffins Pond	3 min	2 Pounds
26	14	Baffins, Portsmouth College	5 min	2 Pounds

Figure 6.1.1. Output without connections

The results of output (Figure 6.1.1) illustrate the direct bus route to the destination. From Hard Interchange to Baffins, Portsmouth College, the traveller can take bus No. 14 directly. The expected time can be shown as well, it is about 27 minutes, while the bus ticket price is 2 pounds. It is more convenient for the traveller to plan their tour.

6.1.2 Output with connections

Route 2 – From South Parade Pier to Baffins Pond by bus No. 6A and 14.

9	6A	Hard Interchange	3 min	2 Pounds
10	6A	Palmerston Road	7 min	2 Pounds
11	6A	South Parade Pier	4 min	2 Pounds
12	6A	Eastney, Health Centre	5 min	2 Pounds
13	6A	St. Mary's Hospital	9 min	2 Pounds
14	6A	Copnor Bridge	6 min	2 Pounds
15	6A	North End, Shops	8 min	2 Pounds
16	6A	Hilsea	6 min	2 Pounds
21	14	Hard Interchange	3 min	2 Pounds
22	14	City Centre	5 min	2 Pounds
23	14	Lake Road/Kingston Road	7 min	2 Pounds
24	14	Copnor Bridge	4 min	2 Pounds
25	14	Baffins Pond	3 min	2 Pounds
26	14	Baffins, Portsmouth College	5 min	2 Pounds

Figure 6.1.2. Output with connections

The results of output (Figure 6.1.2) illustrate the transfer travel plan for users. From South Parade Pier to Baffins Pond, the traveller needs to take bus No. 6A at first, and then transfer to bus No. 14 at the bus stop of Hard Interchange. It also shows the bus stops passing by, which can let traveller know clearly about the bus line route information. The traveller can follow the bus transfer plan suggestion to plan their travel.

6.2 Future Development

This system can realize the Portsmouth city bus route inquiry, including bus number, bus timetable, and bus ticket price. Although at the present time the functions of this system are too simple, the performance of transfer inquiries remains to be further improved and refined, the design and development of the system has a good maintainability and expansibility. With the passage of the time and the system design tools, as well as the development of the environment, the operation of the system can be further strengthened and perfected, eventually to reach an ideal level.

6.2.1 The Problems

This system needs the modules, because of the inadequate ability. When the user finds the bus line information, the system can not provide the print function. The administrator login has not been completed either. The design of database is also small,

only four tables are completed if more data needs to be added, this will result in more complicated tables and, at the same time, in inconvenience to maintain the system. Web queries are operated by the server. If a large number of users are making queries, the operation of server will be affected and in the worse situation the query service will be unsuccessful [7].

The intelligence of the system is not enough. The algorithm of transfer inquiries is to follow with others. This algorithm is close to the original way of thinking, and it is easy to understand. But there is not any form of optimization. Therefore, now the efficiency of the query is lower, and the design is without the distance statistics between two stations. The system only calculates how many stations are passing by. So it can not calculate the shortest path from start station to end station bus this does not mean that the more stations are, the further distance is. On the other hand, the fewer number of stations may be a longer route. All of the above need to be improved in the future [6].

6.2.2 The Development

There are many ways in which this project could be further developed, some of these would give more benefit to the end user, and others could be used to provide a commercial gain.

Firstly, the system needs to deal with the problems which are mentioned above, and then it may be developed as required. The whole objective of this project can be realized, however, the specific modules will need to be improved.

Secondly, the system can be considered with the mobile version. Then the user can use the mobile phone to query the bus information. Nowadays, people prefer to use their mobile phone to surf the internet when they travel. If the system is developed as mobile version as well, it will be useful for more users.

Moreover, in order to bring convenience to traveller who just wants a tour of the city, the system also needs to include information about the places of historical interest and scenic beauty. It may be a useful way to highlight these places of interest, at the same time, and suggestion for travel guide can be provided, as well.

7 Conclusion

With the rapid development of science and technology, computers have been widely used. Nowadays, every company or department can not be operated without computers, which is the reason why computers are used widely. Due to the advantages of computer, it can be simpler and more convenient to have a database in different places, such as management, application, service, or others. At the same time, computers can also improve the efficiency of operations significantly. For this individual project, the design of Portsmouth city bus route online information system is aiming to help users check their travel route more easily.

During the period of the system design, I have learned a new language of programming, as well as building a database with phpMyAdmin. Due to this project, I deeply realize that a system designer needs not only to be good at programming language, but also familiar with the programming tools. At the same time, it can improve my logistical skills. Obviously, when I was doing this individual project, I encountered a lot of problems, and there are many drawbacks for this project. Consequently, this experience will be enhancing and reinforcing in my future study.

REFERENCES

[1] Benjamin, F. & Charles, E. 1998. Shortest Path Algorithms: An Evaluation Using Real Road Networks. Transportation Science. Consulted 1.5.2012 ftp://ftp.etsii.ull.es/pub/asignas/asignas/PRGCOM/trabajos/caminos_min/shortest_path_algorithms.pdf

[2] Han, J. & Chen, X. 2010. Database Design and Implementation. Beijing: Tsinghua University Press. Consulted 1.5.2012 http://www.docin.com/p-264809917.html.

[3] Kofler, M. 2006. The Definitive Guide to MySQL. Transl. Yang, X. Third Edition. Beijing: Post & Telecom Press. Consulted 5.5.2012 http://www.docin.com/p-384885446.html.

[4] Doctor, M. 2010. Beginning Web Programming with HTML, XHTML, and CSS. Transl. Du, J. Second Edition. Beijing: Tsinghua University Press. Consulted 5.5.2012 http://www.docin.com/p-414086667.html.

[5] Wen, Y.; Zhu, Z. & Jia, Z. 2008. PHP5 + MYSQL Web Development Basis and Practice. Beijing: Tsinghua University Press.

[6] Ruan, W. 2010. Javascript Programming Design Tutorial. Second Edition. Beijing: Post & Telecom Press.

[7] Chen, P. 1976. The Entity-Relationship Model: Toward a Unified View of Data. Consulted 15.5.2012 http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.123.1085.

[8] Zhou, H. 2009. Learn Database for 21 days. Shanghai: Electronic Industry Press.

[9] Jiang, P. & Wang, X. 2007. Web System Design. Beijing: Tsinghua University Press.

[10] Chen, W. & Xiao, J. 2008. City Bus Route Query System Algorithm Design and Implementation. Beijing: High Education and Correspondence Education Newspaper.

[11] Chen, Y.; Lu, X. & Di, H. 2006. Optimized Strategy Research over Algorithm of Dijkstra. Computer Technology and Development. Consulted 20.5.2012 http://www.docin.com/p-179869506.html

[12] Philippe, C. 2008. How Database and XML Can Be Used To Master UML[J]. Vol. 23, 220-228.

[13] “PHP W3school online course”. W3school. Consulted 15.3.2012 http://www.w3school.com.cn/php

Appendix

The System Code Implementation

```
<form  method="post" action="default.php">
      <b>From: </b>
<select id="fromaddress1"  name="fromaddress">
<?php
$connect=MySQL_connect('localhost','root','123') or die ("!error connect");
MySQL_select_db("bus_search",$connect);// Connect to MySQL database
$query =MySQL_query("select * from stationinfo  ",$connect);//Get station information
while ($array=MYSQL_fetch_row($query))// To display the search list
{// Start to circulate
                echo " <option value='$array[4]'>$array[1]</option>";//Get station list
}
?>
</select> <br>
        To:   
<select id="toaddress"   name="toaddress">
<?php
$query =MySQL_query("select * from stationinfo  ",$connect);
while ($array=MYSQL_fetch_row($query))
{
                echo " <option value='$array[4]'>$array[1]</option>";
}
?>
</select>
```

```
                                        <br>
                   <input type="submit"
value="Search" />
                                        </form>
    <?php
                        $fromaddress=$_POST['fromaddress'];
                        $toaddress=($_POST['toaddress']);
                        ?>
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no"/>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>BUS</title>

<script                                                 type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
  var directionDisplay;
  var directionsService = new google.maps.DirectionsService();
  var map;
  var haight = new google.maps.LatLng(<?php echo $fromaddress ?>); //Get the
location XY of start station
  var oceanBeach = new google.maps.LatLng(<?php echo $toaddress ?>);//Get the
location XY of end station.
```

```
function initialize() { //Google Map initialize.
  directionsDisplay = new google.maps.DirectionsRenderer();
  var myOptions = {
    zoom: 14,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    center: haight
  }
  map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
  directionsDisplay.setMap(map);
   calcRoute();
}
function calcRoute() {
  var selectedMode ="DRIVING";// document.getElementById("mode").value;
  var request = {
      origin: haight,
      destination: oceanBeach,
      // Note that Javascript allows us to access the constant
      // using square brackets and a string value as its
      // "property."
      travelMode: google.maps.DirectionsTravelMode[selectedMode]
  };
  directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
```

```
        directionsDisplay.setDirections(response);
      }
    });
  }
</script>
<style>
html, body {
  height: 650px;
  margin: 0;
  padding: 0;
}
#map_canvas {
  height: 500px;
  width:650px;
}
@media print {
  html, body {
    height: 650px;
  }
  #map_canvas {
    height: 650px;
  }
}
```

```
</style>
</head>
<body onload="initialize()">
<div>
<b>Bus Travel Plan Suggestion: </b>
</div>
<div id="vb">
  <table width="100%" border="1">
    <tr>
      <td width="500" valign="top"><div id="map_canvas"></div></td>
        <td align="left" valign="top">
            <?php
if( $_POST )
{
$myquery=MYSQL_query("select * from stationinfo where XY='{$_POST['fromaddress']}'",$connect); //Through X Y longitude and latitude to locate the certain station.
if(@mysql_num_rows($myquery)>0)//Error tolerance,  processing, data executing.
{
$fromaddressName= mysql_result($myquery,0,"StationName");//Define a variable value, save start station name.
}
$myEndquery=MYSQL_query("select * from stationinfo where XY='{$_POST['toaddress']}'",$connect);
if(@mysql_num_rows($myEndquery)>0)
```

```
{
$ToaddressName= mysql_result($myEndquery,0,"StationName");//Define a variable value, save end station name.
}
$myEndqueryForLine=MYSQL_query("SELECT a.busid
FROM (
SELECT *
FROM bus_station
WHERE StationName = '{$fromaddressName}'
)a, (
SELECT *
FROM bus_station
WHERE StationName = '{$ToaddressName}'
)b
WHERE a.BusID = b.BusID ",$connect);//Get bus line name.
if(@mysql_num_rows($myEndqueryForLine)>0)
{
$BusID= mysql_result($myEndqueryForLine,0,"BusID");//Define a variable value, save bus line name.
}else
{$BusID= "Prompt:NO data!";}
echo "Depart:";
echo mysql_result($myquery,0,"StationName");
echo "<br> Arrive:  ";
```

```
echo mysql_result($myEndquery,0,"StationName");

echo "<br> Take Bus No.";

echo $BusID; //Display bus line name.

echo " Line.<br>";

$myEndquery=MYSQL_query("SELECT SUM(runtime) AS runtime FROM bus_station WHERE busid='{$BusID}' ORDER BY sn asc",$connect);//Get bus run time.

if(@mysql_num_rows($myEndquery)>0)

{

$runtime= mysql_result($myEndquery,0,"runtime");

}

$ticketPricequery=MYSQL_query("SELECT ticketPrice FROM bus_station WHERE busid='{$BusID}' ORDER BY sn asc limit 1

",$connect);//Get ticket price, through bus ID.

if(@mysql_num_rows($ticketPricequery)>0)

{

$ticketPrice= mysql_result($ticketPricequery,0,"ticketPrice");

}

echo "<br> Expected Time: {$runtime} min

    <br> Ticket Price：{$ticketPrice} Pounds <br>";

?>

      <br><b>Bus No.<?php

      echo $BusID;

                              ?> Stations Information: </b>

      <table width="98%" border="1">
```

```
<tr>
  <td align="center" bgcolor="#F0F8FF"><strong>ID</strong></td>
   <td align="center" bgcolor="#F0F8FF"><strong>Bus No.</strong></td>
  <td align="center" bgcolor="#F0F8FF"><strong>Station Name</strong></td>
 <td align="center" bgcolor="#F0F8FF"><strong>Run Time</strong></td>
   <td align="center" bgcolor="#F0F8FF"><strong>Ticket Price</strong></td>
</tr>
<?php
$query =MySQL_query("SELECT * FROM bus_station WHERE busid='{$BusID}' ORDER BY sn asc",$connect);//Get all station names, according to bus ID.
if(@mysql_num_rows($query)>0)
{
while ($array=MYSQL_fetch_row($query))//Circulate to get the bus station information.
{
?>
<tr>
  <td><?php
                                    echo "$array[0]";//Circulate to display the field.
                                    ?>
                                    </td>
   <td><?php
                                    echo "$array[1]";
                                    ?>
                                    </td>
```

```
<td><?php
                                    echo "$array[2]";
                                    ?></td>
<td><?php
                                    echo "$array[4]";
                                    ?> min</td>
<td><?php
                                    echo "$array[5]";
                                    ?> Pounds</td>
</tr>
<?php
                        }}
                        ?>
</table>
<?php
                        $queryFortransfer=MYSQL_query("select distinct
a.stationName from
(
select * from  `bus_station` where busid in(
SELECT
 `BusID`
FROM
 `bus_station` where stationName='{$fromaddressName}' or
stationName='{$ToaddressName}')
```

```
 ) a,
 (
select * from  `bus_station` where busid in(
SELECT
 b1.`BusID`
FROM
 `bus_station` b1, `bus_station` b2
 where b1.busid = b2.busid and b1.stationName='{$fromaddressName}' and b2.stationName='{$ToaddressName}')
 ) b where a.stationName=b.stationName and a.busid!=b.busid",$connect);
if(@mysql_num_rows($queryFortransfer)>0)
{
$transfer= mysql_result($queryFortransfer,0,"stationName");
}else
{$transfer= "Prompt:No data!";}
                                    ?>
      Transfer Plan:<br/>From <?php
                                                echo $fromaddressName;
                                                ?> to <?php
                                                echo $ToaddressName;
                                                ?> at <?php
                                                echo $transfer;
                                                ?> transfer.
```

```
<table width="98%" border="1">
<tr>
<td align="center" bgcolor="#F0F8FF"><strong>ID</strong></td>
<td align="center" bgcolor="#F0F8FF"><strong>Bus No.</strong></td>
<td align="center" bgcolor="#F0F8FF"><strong>Station Name</strong></td>
<td align="center" bgcolor="#F0F8FF"><strong>Run Time</strong></td>
<td align="center" bgcolor="#F0F8FF"><strong>Ticket Price</strong></td>
</tr>
<?php
$Forquerytransfer =MySQL_query("select * from  `bus_station` where busid in(
SELECT
 b1.`BusID`
FROM
 `bus_station` b1, `bus_station` b2
 where b1.busid = b2.busid and b1.stationName='{$fromaddressName}' and b2.stationName='{$ToaddressName}') ORDER BY id asc",$connect);
 //Get transfer information
if(@mysql_num_rows($Forquerytransfer)>0)
{
while ($array=MYSQL_fetch_row($Forquerytransfer))//Get station information of transfer bus.
{
?>
<tr>
```

```
<td><?php
                                    echo "$array[0]";
                                    ?>
                                    </td>
 <td><?php
                                    echo "$array[1]";
                                    ?>
                                    </td>
<td><?php
                                    echo "$array[2]";
                                    ?></td>
<td><?php
                                    echo "$array[4]";
                                    ?> min</td>
<td><?php
                                    echo "$array[5]";
                                    ?> Pounds</td>
</tr>
 <?php
                        }}
                        ?>
</table>
<?php
```

```
                    }
                    ?>
        </td>
    </tr>
    <tr>
     <td colspan="2" align="center">Shaoyu Wang 2012 - Produced as part of the individual project Assignment</td>
    </tr>
 </table>
</div>



</body>
</html>
```